1. Что такое DOM? Зачем он нужен?
2. Расскажите о возможностях JavaScript
3. Расскажите об операторах var, let, const, их различиях и областях видимости.
4. Назовите несколько рекомендаций по стилю кода, как должны именоваться переменные?
5. Перечислите основные типы данных в JavaScript. Как узнать тип данных переменной?
6. История создания JavaScript, спецификации
7. Как создавать и использовать функции? Что такое стрелочные функции?
8. Что такое объекты в JavaScript, свойства объектов.
9. Расскажите о JSON, как и где используется.
10. Тернарный оператор в JavaScript, пример использования
11. Назовите 5 функций для работы с массивами, примеры использования
12. Что такое XMLHttpRequest, для чего нужен?
13. Promise – что это и для чего нужно?
14. Что такое fetch?
15. Async, await?
16. Назовите 5 браузерных событий
17. Регулярные выражения
18. Что такое LocalStorage, как записать и получить значение?
19. Что такое Express.js?
20. Что такое Node.js?
21. Что такое MongoDB?
22. Что подразумевает собой клиент-серверная архитектура приложений?

Практические:

1. Получить с адреса «https://jsonplaceholder.typicode.com/posts/54» пост с id 54 и вывести его на страницу, используя следующий макет:



К этому посту вывести комментари с https://jsonplaceholder.typicode.com/posts/54/comments, содержащие: имя и текст комментария.

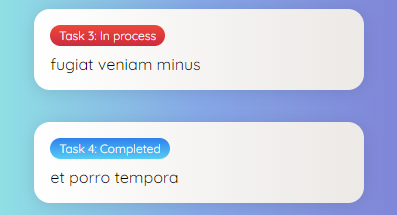
Стилизовать комментарии на свое усмотрение.

1. Получить с адреса «https://jsonplaceholder.typicode.com/todos» список задач. Вывести 25 первых задач на страницу.

Требования:

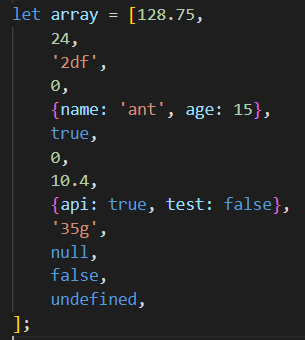
* Необходимо пронумеровать задачи
* Задний фон номера задачи должен быть синий, либо красный в зависимости от того выполнена ли задача. Также необходимо менять содержание блока, если задача выполнено – «Completed», иначе «In process»

Необходимо придерживаться следующей разметки:

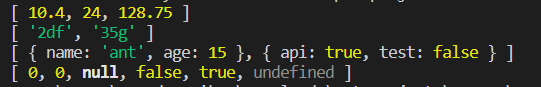


* Для вывода задач использовать display: grid.
* Задачи должны выглядеть красиво и отличаться друг от друга только фоном (если задача выполнена – лаймовый, если нет – белый).

1. Пусть дан массив вида:



Создать функцию, которая принимает в качестве агрумента – массив, сортирует элементы и создает новые массивы для каждого типа данных находящихся в исходном массиве (исключая нули, null, undefined и boolean – их поместить в отдельный массив для мусора). Вывести все получившиеся массивы в консоль.



1. Напишите функцию getArrayOfRange(), принимающую два аргумента: начало и конец диапазона, и возвращающую массив, который содержит все числа из диапазона, включая начальное и конечное. Третий необязательный аргумент функции – шаг для построения массива.

Убедитесь, что функция работает с отрицательным шагом: например, getArrayOfRange(5, 2, -1) возвращает [5, 4, 3, 2].

На html-странице определить поля ввода начального, конечного значения и шага, после выполнения вывести результат на страницу.

1. Создать функцию, которая выведет количество гласных и согласных в строке (латиница). Использовать регулярные выражения.

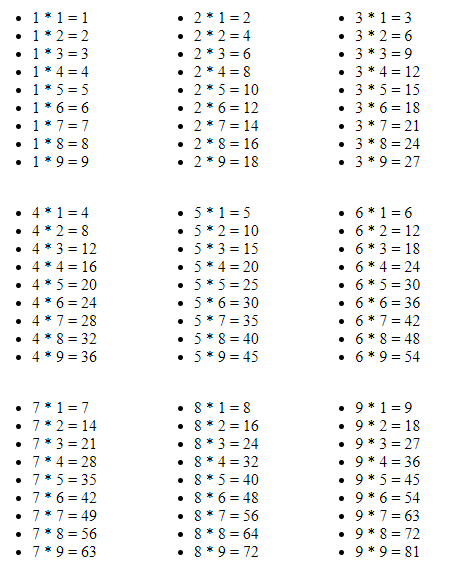
После определить на странице поле ввода, под ним отображать текущее кол-во гласных и согласных. (Значения должны обновляться при каждом изменении в поле ввода)

1. Напишите функцию, которая выводит на страницу элемент – число от 1 до 100, с двумя исключениями.

* Для чисел, нацело делящихся на N, она должна выводить текущее число и текст ‘Fizz’ с оранжевым фоном
* Для чисел, делящихся на K (но не на N) – число и текст ‘Buzz’ с красным фоном.

Когда сумеете – исправьте её так, чтобы она выводила число и текст «FizzBuzz» с зеленым фоном для всех чисел, которые делятся и на N, и на K. **Значения N и K должны вводиться пользователем.**

1. Напишите функцию, которая создает на странице таблицу умножения.



1. Определить на странице 2 поля ввода «Title», «Body» и кнопку «Send», по нажатию на кнопку отправить POST-запрос по адресу «https://jsonplaceholder.typicode.com/posts», после выполнения запроса вывести на страницу созданный пост вместе с id.

* При выводе на страницу нового поста использовать ответ jsonplaceholder.
* Стилизовать пост используя макет ниже.

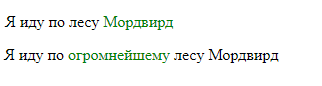


1. Написать функцию, которая возвращает значение true, если строка является палиндромом, и false — если нет.

Палиндром — слово, предложение или последовательность символов, которая абсолютно одинаково читается как в привычном направлении, так и в обратном. К примеру, “Anna” — это палиндром, а “table” и “John” — нет.

1. Написать функцию, которая возвращает самое длинное слово в предложении.

Предложение должен вводить пользователь, после нажатия кнопки «Найти» на страницу выведется предложение с выделенным зеленым цветом словом



Темы для повторения:

1. История создания JavaScript, возможности языка, область применения, особенности, спецификации.
2. DOM, основные события, работа с window.
3. Var, let, const. Области видимости и различия.
4. Функции и способы их объявления.
5. Объекты, свойства объектов, методы работы с ними.
6. XMLHttpRequest и Fetch, синтаксис, использование.
7. Суть Promise и пример применения.
8. Async – await, определения и использование.
9. Объявление и использование регулярных выражений, флаги.
10. Node + Express, использование для создания backend, ответы на запросы.
11. NPM – инициализация проекта, установка пакетов.